



# The Unified Modeling Language

# UML: активности



# UML



Г. Буч, А. Якобсон, Дж. Рамбо UML  
Издание второе Издательство: Питер  
Серия: Классика Computer Science 2006  
г., с. 736

Эта книга представляет собой полный справочник по языку UML. Она адресована в первую очередь разработчикам, системным архитекторам, руководителям проектов, инженерам-системщикам, программистам, аналитикам, заказчикам и вообще всем, кому по роду деятельности приходится описывать, проектировать и строить сложные программные системы, а также разбираться в их функционировании. В книге дается всестороннее описание понятий и конструкций UML, включая их семантику, нотацию и назначение. Материал организован таким образом, чтобы книгой было удобно пользоваться, несмотря на ее объем и полноту содержания.

# UML



Дж. Рамбо, М. Блаха *UML 2.0. Объектно-ориентированное моделирование и разработка:* Издательство: Питер, Год: 2007, 544 с.

Новое издание обновлено в соответствии со стандартом UML 2.0. Авторы четко и ясно объясняют суть важнейших концепций объектно-ориентированного программирования, представляют способы реализации этих идей при разработке ПО с использованием языков C++ и Java, а также реляционных баз данных. В книге есть задания и множество советов, что делает ее очень практичной.

# UML



**Джим Арлоу, Айла Нейштадт *UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование*: Издательство: Символ-Плюс, 2007, С. 624**

Практическое руководство по сложному процессу объектно-ориентированного анализа и проектирования с помощью UML 2. В книге показано место ОО анализа и проектирования в цикле разработки программного обеспечения, как его определяет Унифицированный процесс (UP)..

# UML



**А. Леоненков Самоучитель UML. Эффективный инструмент моделирования информационных систем: Издательство: BHV - Санкт – Петербург, 2001, с. 304**

В книге рассматриваются основы UML - унифицированного языка моделирования для описания, визуализации и документирования объектно-ориентированных систем и бизнес-процессов в ходе разработки программных приложений. Подробно описываются базовые понятия UML, необходимые для построения объектно-ориентированной модели системы с использованием графической нотации. Изложение сопровождается примерами разработки отдельных диаграмм, которые необходимы для представления информационной модели системы.

# Основные разработчики языка UML (Three amigos)



Grady Booch  
Гради Буч



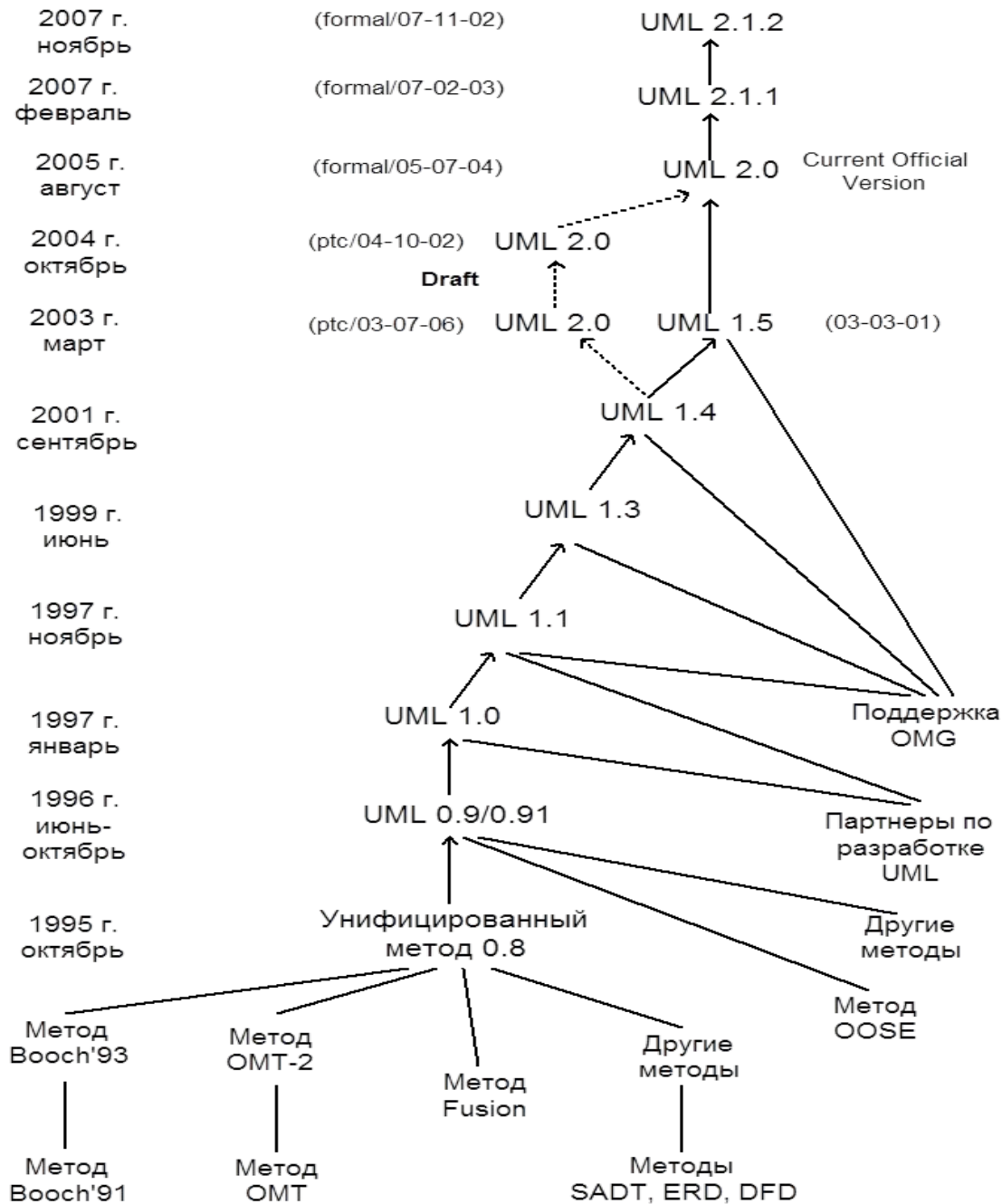
Dr. James  
Rumbaugh  
Джеймс Рамбо  
(Джим Румбах)



Dr. Ivar  
Jacobson  
Айвар  
Джекобсон

- **OMG** (Object Management Group) — название консорциума, созданного в 1989 году для разработки промышленных стандартов с их последующим использованием в процессе создания масштабируемых неоднородных распределенных объектных сред.
- Официальный сайт: [www.omg.org](http://www.omg.org)

# История развития языка UML





# Определение языка UML

*Unified Modeling Language* — унифицированный язык моделирования для описания, визуализации и документирования объектно-ориентированных систем в процессе их анализа и проектирования

**Язык UML предоставляет стандартный способ написания проектной документации на системы**

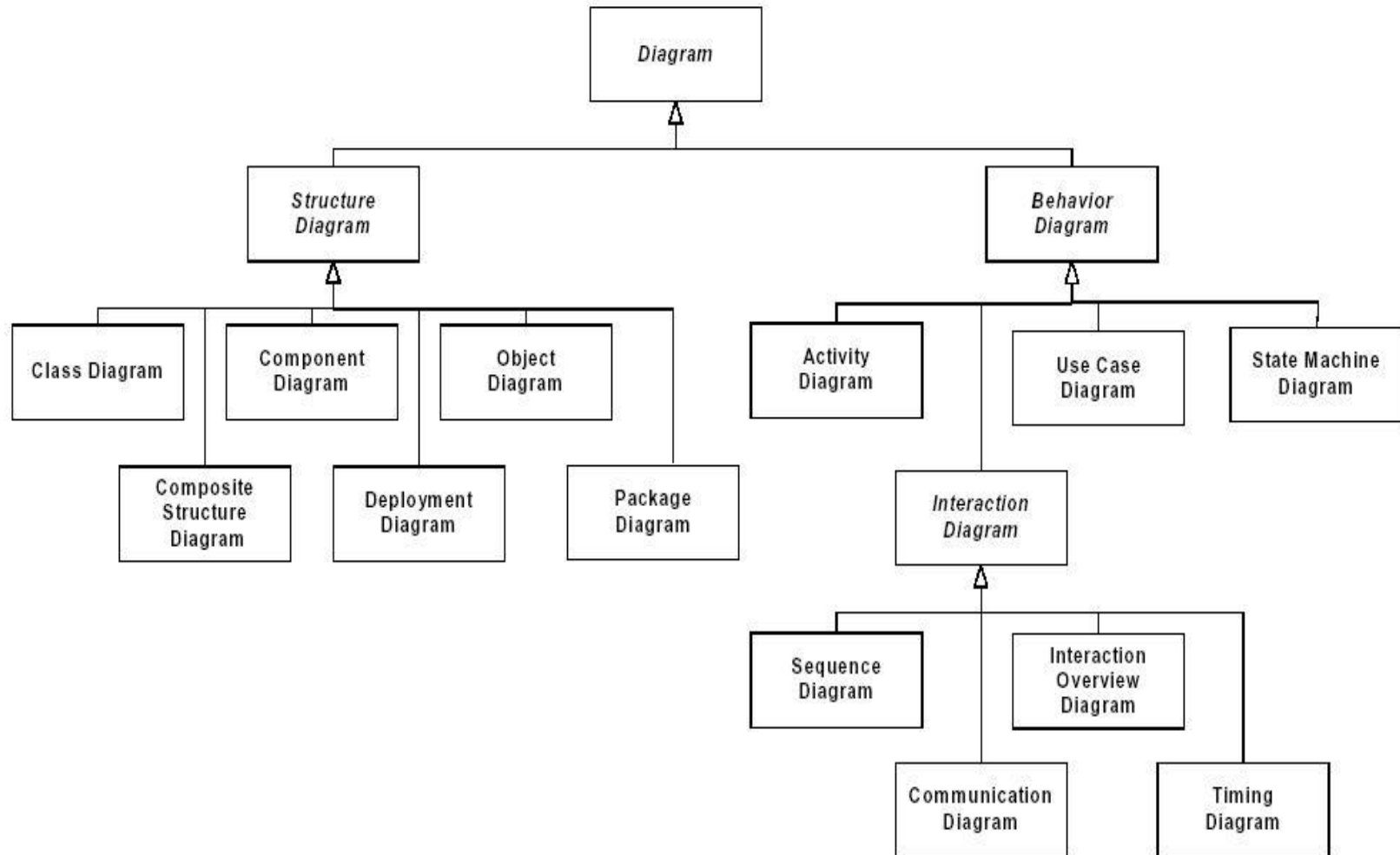
# Назначение языка UML

- Предоставить разработчикам легко воспринимаемый и выразительный язык визуального моделирования, специально предназначенный для разработки и документирования моделей сложных систем различного целевого назначения
- Снабдить исходные понятия языка UML возможностью расширения и специализации для более точного представления моделей систем в конкретной предметной области
- Графическое представление моделей в нотации UML не должно зависеть от конкретных языков программирования и инструментальных средств проектирования
- Описание языка UML должно включать в себя семантический базис для понимания общих особенностей ООАП
- Способствовать распространению объектных технологий и поощрять развитие рынка программных инструментальных средств
- Интегрировать в себя новейшие и наилучшие достижения практики ООАП

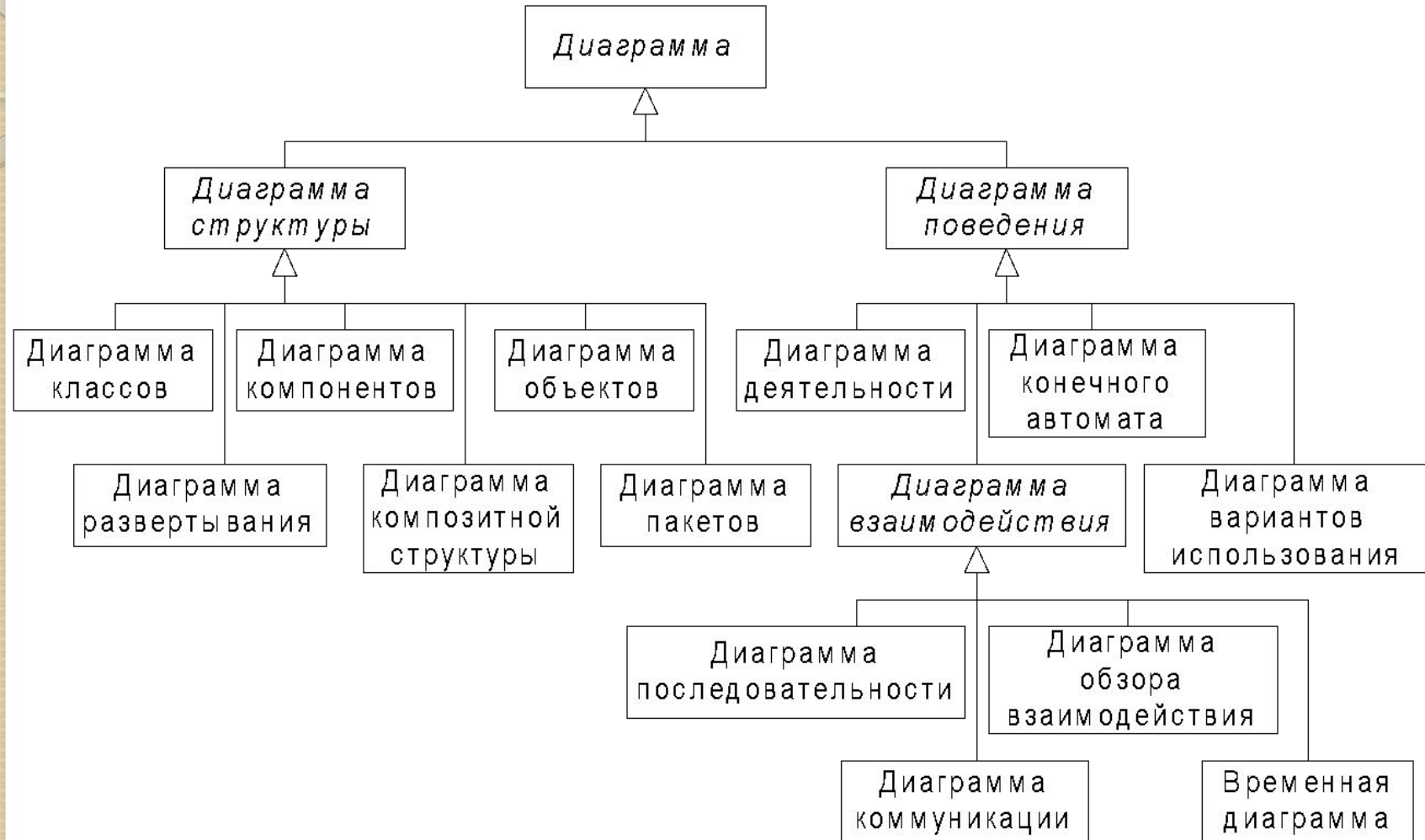
# Классификация моделей в языке UML

- *Структурные модели (structured models)* – модели, предназначенные для описания статической структуры сущностей или элементов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения.
- *Модели поведения (behavioral models)* – модели, предназначенные для описания процесса функционирования элементов системы, включая их методы и взаимодействие между ними, а также процесс изменения состояний отдельных элементов и

# Канонические диаграммы языка UML 2.x



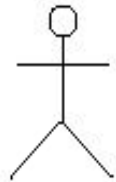
# Канонические диаграммы языка UML 2.x



# Назначение диаграммы вариантов использования

- Определить общие границы функциональности проектируемой системы в контексте моделируемой предметной области.
- Специфицировать требования к функциональному поведению проектируемой системы в форме вариантов использования.
- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями

# Основные обозначения на диаграмме вариантов использования



actor



use case



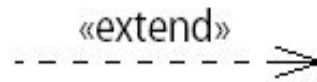
system boundary



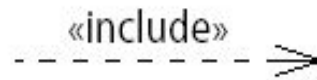
communication association



generalization



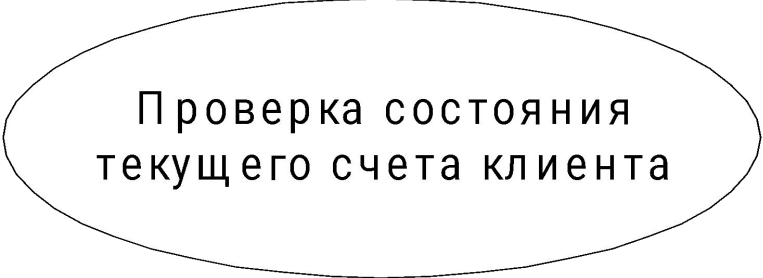
extend



include

# Вариант использования (use case)

- Представляет собой общую спецификацию совокупности выполняемых системой действий с целью предоставления некоторого наблюдаемого результата, который имеет значение для одного или нескольких актеров
- Отвечает на вопрос «Что должна выполнять система?», не отвечая на вопрос «Как она должна выполнять это?»
- Имена – отглагольное существительное или глагол в неопределенной форме

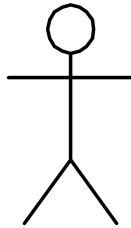


Проверка состояния  
текущего счета клиента



# Актер (actor)

- Любая внешняя по отношению к проектируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач
- *Примеры актеров*: кассир, клиент банка, банковский служащий, президент, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон



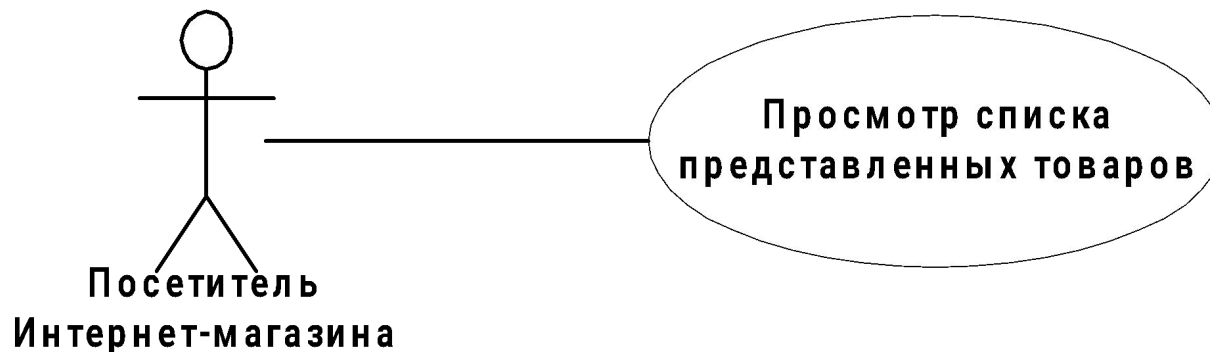
Клиент банка

# Вопросы для идентификации актеров в системе

- Какие организации или лица будут использовать систему
- Кто будет получать пользу от использования системы
- Кто будет использовать информацию от системы
- Будет ли использовать система внешние ресурсы
- Может ли один пользователь играть несколько ролей при взаимодействии с системой
- Могут ли различные пользователи играть одну роль при взаимодействии с системой
- Будет ли система взаимодействовать с законодательными, исполнительными, налоговыми или другими органами

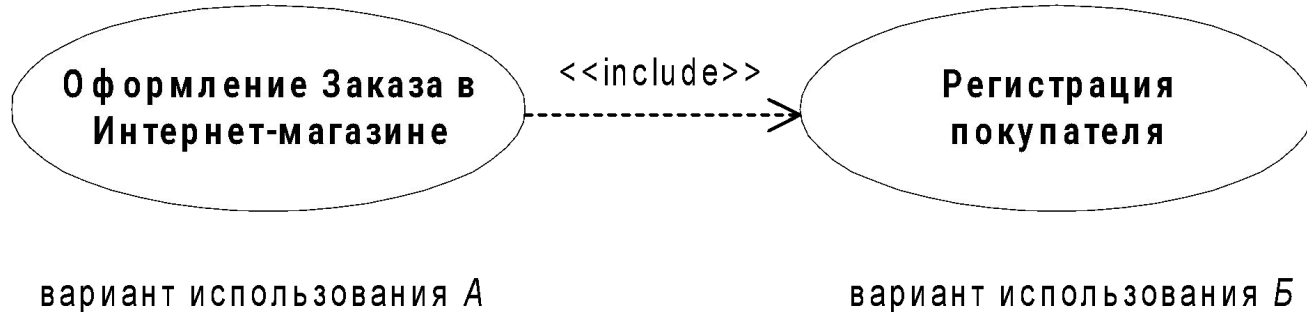
# Отношение ассоциации

- Ассоциация (association) является одним из фундаментальных понятий в языке UML 2.x и может использоваться на различных канонических диаграммах при построении визуальных моделей
- Применительно к диаграммам вариантов использования отношение ассоциации может служить только для обозначения взаимодействия актера с вариантом использования.



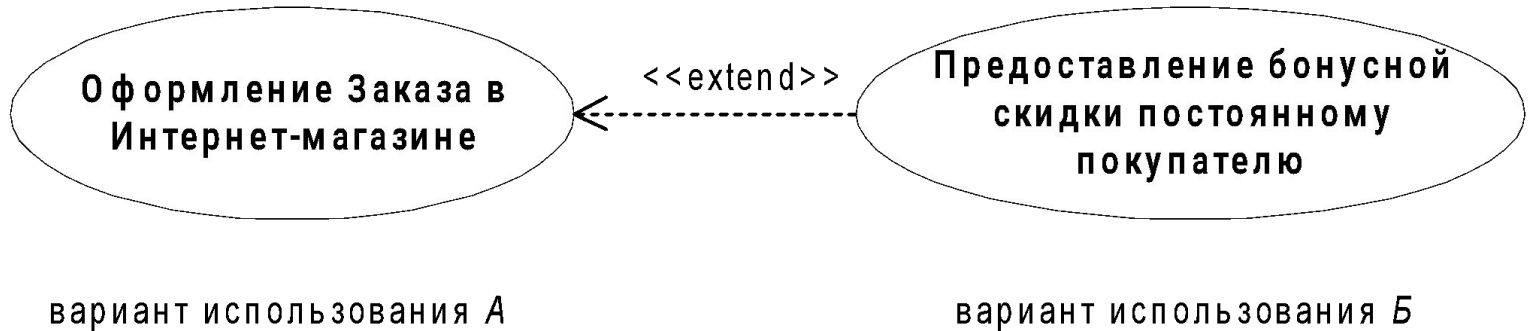
# Отношение включения

- Отношение *включения (include)* специфицирует тот факт, что некоторый вариант использования содержит поведение, определенное в другом варианте использования



# Отношение расширения

- Отношение *расширения* (*extend*) определяет взаимосвязь одного варианта использования с некоторым другим вариантом использования, функциональность или поведение которого задействуется первым не всегда, а только при выполнении некоторых дополнительных условий.



# Отношение обобщения

- *Отношение обобщения (generalization relationship)* предназначено для спецификации того факта, что один элемент модели является специальным или частным случаем другого элемента модели

